# An Innovative Network based on Double Receptive Field and Recursive Bi-directional Long Short-Term Memory

**Peng-fei Meng**
Mogo Auto Intelligence and Telematics Information Technology Co., Ltd
**Shuang-cheng Jia**
Mogo Auto Intelligence and Telematics Information Technology Co., Ltd
**Qian Li** ( liqian@mogoauto.com )
Mogo Auto Intelligence and Telematics Information Technology Co., Ltd

**Research Article**

**Keywords:** Character recognition, Neural network, Residual, BiLSTM, OCR, CRNN, Convolution, Receptive field

**Posted Date:** June 4th, 2021

**DOI:** https://doi.org/10.21203/rs.3.rs-579378/v1

# An innovative network based on double receptive field and Recursive Bi-directional Long Short-Term Memory

*Peng-fei Meng [1], Shuang-cheng Jia [1], and Qian Li [1,*]*

*[1]Autonomous Driving Department, Mogo Auto Intelligence and Telematics Information Technology Co., Ltd,*

**Corresponding author，liqian@mogoauto.com. Address: 36 Beisanhuan Dong Road, Dongcheng District, Beijing 100013, China*

***Abstract*—In the scenes of character recognition, this paper studies the influence of network structure and parameters of "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Re cognition (CRNN)" on the recognition results in detail, and proposes an improved CRNN network based on double receptive field and recursive Bi-directional Long Short-Term Memory（BILSTM）, which is named "An innovative network based on double receptive field and Recursive Bi-directional Long Short-Term Memory（CRNN_RES）". In the CRNN_RES network, the innovations of this paper are adjusting the structure of CNN to enhance the feature extraction ability of the CNN network and using the shared parameter BILSTM network with recursive residuals to reduce the number of network parameters and improve the accuracy of the model prediction. In fact, the number of parameters of CRNN_RES network proposed in this paper is 7148325, which is 1182976 fewer than that of CRNN. On the same open datasets:ICDAR 2003（IC03），ICDAR 2013（IC13），IIIT 5k-word（IIIT5k）, and Street View Text (SVT), the proposed method achieves 96.90%, 89.85%, 83.63%, and 82.96% recognition accuracy, which are higher than that of CRNN 1.40%, 3.15%, 5.43%，and 2.16%.**

***Index Terms*—Character recognition, Neural network, Residual, BiLSTM, OCR, CRNN, Convolution, Receptive field.**

## I. INTRODUCTION

Character recognition methods are mainly divided into two categories: recognized by traditional algorithms or recognized by neural network algorithm. If we use the traditional algorithm to recognize, when we design the algorithm, it is necessary to consider more detailed image features to process the image noise, image quality, resolution, etc. With the development of the neural network, the neural network had been widely used in image recognition technology, and then image recognition technology achieved a major breakthrough. At present, the Computational Vision (CV) world is undoubtedly the world of neural networks. In 1985, Rumelhart and Hinton proposed a BP neural network[1], which made the training of neural network much easier. In 1988, LeCun proposed the Lenet5 network[2], Lenet5 used a convolutional neural network (CNN) for the first time, using convolution, pooling, and nonlinear three layers as a series. In 2010, Dan Claudiu Ciresan and Jurgen Schmidhuber invented a neural network that can be trained on Graphics Processing Unit (GPU)[3]. In 2012, Alex Krizhevsky invented the AlexNet deep neural network[4] and won the champion of the ImageNet competition. Subsequently, more and deeper neural networks were invented. In 2014, the University of Oxford invented the Visual Geometry Group (VGG) network[5], and used the small convolution kernel of 3*3 in each convolution layer for the first time. In ImageNet Large Scale Visual Recognition Challenge (ILSVRC) localization and classification competition, VGG obtained first and second place respectively. In 2015, He Kaiming invented the Residual Neural Network (RESNET) deep neural network[6], proposed and applied the concept of residual network, which solved the network degradation problem with the increase of network level, and the neural network became deeper and had better effect. After that, neural network developed rapidly in image recognition, and various networks emerged in an endless stream. At present, CRNN[7] network is widely used in the field of Optical Character Recognition (OCR). The CRNN [7] network is widely used because it uses the network architecture of CNN+RNN(Recurrent Neural Network)+tranpose layer, which makes it possible to recognize the sequence of images. One feature of the sequence object is that its length is variable, and CRNN solves the recognition problem of image sequences, so CRNN has this network architecture that can recognize characters of indefinite length. . It can be trained end-to-end and can meet the recognition of characters with variable length, which is one of the main reasons that CRNN is widely used in the field of pattern recognition. After studying in detail the network structure and calculation principle of CRNN, we

proposed a new and improved CRNN network, in which the last two layers of BILSTM[8,9] of the original CRNN[7] is changed into a layer of BILSTM with shared parameters. Tests were carried out on four kinds of public datasets. Compared with the original CRNN network, the improved model has less model parameters, smaller model size, higher accuracy and faster recognition speed.

## II. THE STRUCTURE OF THE CRNN_RES NETWORK AND THE MODIFICATIONS OF THE RNN LAYER

The overall network structure of CRNN[7] consists of three parts: convolution layer, RNN layer, and transcription layer. Because when inputting data to the Recurrent Neural Network (RNN), there is no need for each element to obtain position information in the sequence, so we are also based on the RNN network when improving the CRNN network, but here we have made many modifications on the RNN. The network architecture proposed in this article is similar to CRNN but there are also big differences. From the RNN part of the network architecture diagram in Fig.1 of the network we proposed (CRNN_RES), we can see that the RNN layer of the CRNN_RES network proposed in this article is significantly different from the RNN layer of CRNN. The RNN part of the network structure proposed in this article is introduced as follows:

We use BiLSTM [8,9] as the basic network of the RNN layer. BiLSTM is a two-way LSTM network, the traditional LSTM can only learn the one-way feature dependency of the image sequence, but the sequence we recognize may have a reverse dependency, such as the words "google", "brother", we can predict the words "google" and "brother" based on "googl" and "brothe", or predict the words "google" and "brother" based on "oogle" and "rother". So here we choose the bidirectional LSTM as the basic network in the RNN module of the CRNN_RES.

CRNN_RES adds a short-circuit connection between the output of the convolutional layer and the output of the BILSTM layer, and the result of the first output is taken as part of the input of the BILSTM. The specific process is as follows: firstly, CONO (the output of the network in the convolutional layer) is put into the BILSTM layer, and then L1(the output result of the BILSTM layer) and CONO (the output result of the convolutional layer) are added to get a result, which is represented by O1 here. The specific process can be expressed as equation 1:

$$O_1 = L_1 + CONO \tag{1}$$

Then we input O1 into BILSTM, and L2(the result of the second BILSTM layer) is added to O1 and CONO (the output of the convolutional layer), and finally, O2(the final output of RNN layer) is obtained, which is expressed as equation 2:

$$O_2 = CONO + O_1 + L_2 \tag{2}$$

The process is shown in Figure 1.

The output of the convolution layer is denoted as CONO, and the operation function of BILSTM is denoted as F, then the first and second operations can be expressed as equation 3:

$$\begin{aligned} O_1 &= CONO + F(CONO) \\ O_2 &= F(O_1) + CONO + O_1 \end{aligned} \tag{3}$$

The specific process of the RNN layer is as follows: we input data which has a shape of [timestep,batchSize,512] firstly to the BILSTM, and we will get an output data which has a shape of [timestep,batchSize,512]. Then we add the previous output of the BILSTM to the output of the convolution layer, and the output shape is [timestep, batchSize, 512]. The previous output result is input to BiLSTM again, the input shape of data is [timestep, batchSize, 512], the output shape is [timestep, batchSize, 512]. The second BiLSTM operation results are then added to the pr evious two outputs, and then the output dimension of the data i s changed by the full connection layer. The final shape of the output is [timestep * batchSize, nclass], finally we convert the output, the final output shape is [timestep, batchSize, nclass]. Here timestep refers to the length of time series, batchSize refers to the number of each batch of pictures input to the network during training, and nclass refers to the number of categories of classification. Figure 1 is the overall architecture diagram of the network proposed in this paper, and Figure 2 is the detailed structure diagram of the RNN part of the network proposed in this paper. Readers can have an intuitive understanding of the overall architecture of the network through figure 1, without going into the details in Figure 1. Then through figure 2 to understand how the RNN part is designed.

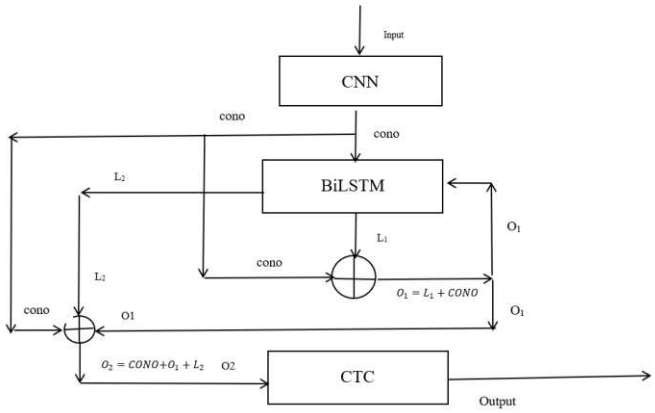



Fig. 1. Summary architecture diagram of the RNN layer in the CRNN_RES network.

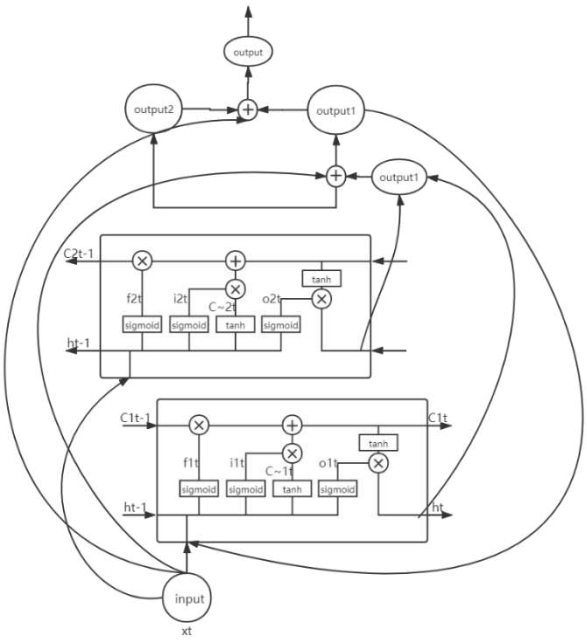



Fig. 2. Detailed architecture diagram of the RNN layer in the CRNN_RES network

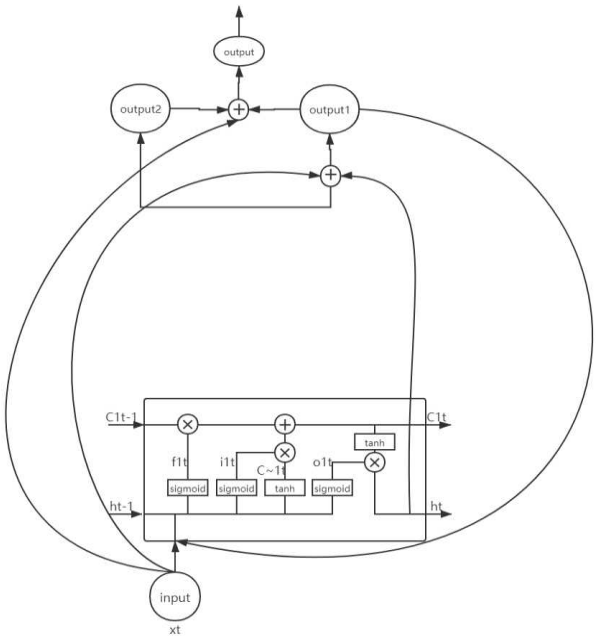



Fig. 3. the form of one-way LSTM.

## III. THE MATHEMATICAL PRINCIPLE OF RNN IN THE CRNN_RES NETWORK

For CRNN_RES networks, the calculation principle is the same as LSTM[9] or BiLSTM. In order to make the elaboration of the calculation principle more concise and convenient for readers to understand, we only calculate the form of one-way LSTM. All mathematical calculations are based on the following network architecture. The following Figure 3 is a simplified form of the RNN part of the network structure proposed in this paper. It represents the network structure diagram when one-way LSTM is used. Our purpose is to make it easier for readers to understand the network structure. Readers can more intuitively understand the RNN part of the network structure proposed in this paper through the following Figure 3.

1) Calculate the information of the forgetting gate:

$$f1t = \text{sigmoid}\ (wf * [ht\text{-}1\ ,xt] + bf) \tag{4}$$

Among them, wf represents the weight, bf represents the bias term, ht-1 is the output of the last hidden unit of lstm, xt is the input of the current hidden neural unit, and we use sigmoid as the activation function.

2) Calculate the information of the memory gate:

$$i1t = \text{sigmoid}\ (wi[ht\text{-}1, xt] + bi) \tag{5}$$

Among them, wi represents the weight, bi represents the bias term. ht-1 is the output of the last hidden unit of lstm, xt is the input of the current hidden neural unit, and we use sigmoid as the activation function.

$$C\sim 1t = \tanh\ (wc[ht\text{-}1\ ,xt] + bc\ ) \tag{6}$$

Among them, wc represents the weight, bc represents the bias term

3) Calculate the neural unit state at the current moment:

$$C1t = C1t\text{-}1 * f1t + i1t * C\sim 1t \tag{7}$$

C1t-1 represents the state of the cell at the previous moment.

4) Calculate the information of the output gate:

$$o1t = \text{sigmoid}\ (wo[ht\text{-}1\ ,xt] + bo) \tag{8}$$

Among them, wo represents the weight, bo represents the bias term

5) Calculates the current state of the current hidden layer:

$$Ht = o1t * \tanh(C1t) \quad (9)$$

C1t represents the state of the cell at the current moment.

At this time, the calculation of LSTM unit is completed.

6) Next, perform the following calculations:

$$output1 = xt + ht \quad (10)$$

xt is the input of the current hidden layer neural unit, ht is the output of the current hidden unit of lstm.

Re-input output1 into the lstm unit.

Then there are:

$$\begin{aligned} f1t &= \text{sigmoid}(wf * [ht\text{-}1, output1] + bf) \\ i1t &= \text{sigmoid}(wi[ht\text{-}1, output1] + bi) \\ C\sim 1t &= \tanh(wc[ht\text{-}1, output1] + bc) \\ C1t &= C1t\text{-}1 * f1t + i1t * C\sim 1t \\ o1t &= \text{sigmoid}(wo[ht\text{-}1, output1] + bo) \\ ht &= o1t * \tanh(C1t) \\ output2 &= xt + ht \end{aligned} \quad (11)$$

ht-1 is the output of the last hidden unit of lstm, wf represents the weight, and bf represents the bias. wi represents the weight, and bi represents the bias. wc represents the weight, bc represents the bias, ht-1 is the output of the last hidden unit of lstm.

C1t-1 represents the state of the cell at the previous moment. wo means weight, bo means bias.

Therefore, the output can be calculated as equation 12

$$output = xt + output2 + output1 \quad (12)$$

## IV. MODIFICATIONS OF THE CONVOLUTIONAL LAYER

The convolutional layer of CRNN_RES is used to extract the feature information of the image, which can be regarded as the feature extraction layer of the network. The purpose of our modification to the convolutional network is to make the network have stronger feature extraction capabilities and ensure that each convolutional layer can extract richer image feature information. As you can see from figure 4 below, compared to the convolutional network of CRNN, we added a layer of BatchNomalization [10] after the third layer of convolution, in order to make the network better to fit the features of the image, reduce the probability that the features are too complex and exceed the fitting ability of the network. After the convolution of the first layer and the second layer, a pooling layer with a kernel size of 1x2 was added respectively, and the output results of the pooling layer with a kernel size of 2x2 were fused with those of the pooling layer with a kernel size of 1x2. The purpose is to allow the network to use multiple receptive fields to extract features, enhance the feature extraction capabilities of the network, and make the network have better feature extraction capabilities for both wide characters and narrow characters. In the last two pooling layers, we added the pooling layer with the kernel size of 3x2 respectively and fused the output results of the pooling layer with the kernel size of 1x2 and the output results of the pooling layer with the kernel size of 3x2. The purpose here is to make the network use multiple receptive fields to extract features so that the network is friendly to both wide characters and narrow characters. The purpose of adding pooled layers is to obtain different receptive fields and more receptive field characteristics. In this way, the network can better extract the features of narrow characters and wide characters, so as to improve the recognition accuracy of characters of different sizes.

In order to facilitate the reader's understanding, we will use algebra and graphs to illustrate our changes to the convolutional layer: Assume that the result of maximum pooled branch 1 is A and the result of maximum pooled branch 2 is M. Then the final result P will be obtained by the pooling layer is:

$$P=A+M \quad (13)$$

Our modifications to the CNN module are shown in the following table 1. Readers can refer to the following table 1 to understand the detailed structure and parameters of the cnn_res network. Through the following table 1, readers can easily understand in detail how we add the BatchNormalization(BN) [10] layer and the reason of modifying the pooling layer. There are no changes to the CNN module except for adding a BatchNormalization layer and pooling layers. The detailed network structure of the CNN module is shown in table 1:

**TABLE 1. NETWORK STRUCTURE OF THE CNN SETS**

| Type | Configurations |
|---|---|
| Transcription | - |
| Rnn-res | Hidden units: 512 |
| BatchNormalization | - |
| Convolution | out_channels: 512, kernel_size: 2, stride: 1, padding: 0 |
| MaxPooling | kernel_size: 1x2, stride: 1x2 / kernel_size: 3x2, stride: 1x2, padding:(1,0) |
| BatchNormalization | - |
| Convolution | out_channels: 512, kernel_size: 3, stride: 1, padding: 1 |
| BatchNormalization | - |
| Convolution | out_channels: 512, kernel_size: 3, stride: 1, padding: 1 |
| MaxPooling | kernel_size: 1x2, stride: 1x2 / kernel_size:3x2 ,stride: 1x2 ,padding:(1,0) |
| Convolution | out_channels: 256, kernel_size: 3, stride: 1, padding: 1 |

| | |
|---|---|
| BatchNormalization | - |
| Convolution | out_channels: 256, kernel_size: 3, stride: 1, padding: 1 |
| MaxPooling | kernel_size: 2, stride: 2 kernel_size: 1x2, stride: 2 |
| Convolution | out_channels: 128, kernel_size: 3, stride: 1, padding: 1 |
| MaxPooling | kernel_size: 2, stride: 2 kernel_size: 1x2, stride: 2 |
| Convolution | out_channels: 64, kernel_size: 3, stride: 1, padding: 1 |
| Input | Wx32 gray images |

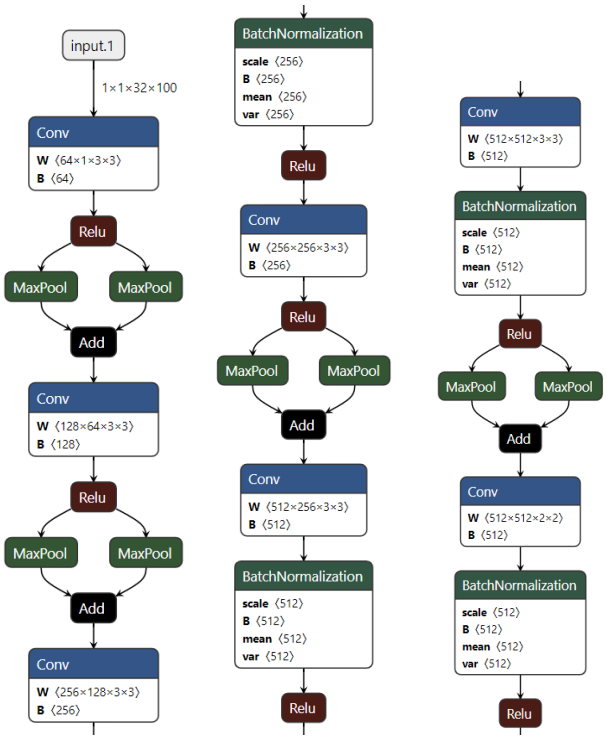



Fig. 4. The CNN layer architecture of CRNN_RES.

## V. EXPERIMENTAL DATA PREPARATION AND HYPERPARAMETERS EXPLANAT

In order to compare with CRNN, and more intuitively show the performance improvement brought by improved network, we chose the same synthetic dataset（Synth）[11] as the training data. The dataset contains 8 million training images and their corresponding actual words. The same dataset as CRNN was selected for testing, that is ICDAR 2003（IC03）[12]，ICDAR 2013（IC13）[13]，IIIT 5k-word（IIIT5k）[14],and Street View Text (SVT)[15], the test set of these four kinds of data, and the partition of the dataset was not modified. The data used is exactly the same as CRNN. Before the images been input into the network, we also scaled the image uniformly and equally to the size of 100x32.

## VI. COMPARISON OF RESULTS

The following are the results calculated using the Tesla V100 GPU.

**TABLE 2. COMPARISON OF RECOGNITION ACCURAY ON DIFFERENT DATA**

| network dataset | IC03 | IC13 | IIIT5k | SVT |
|---|---|---|---|---|
| CRNN | 0.894 | 0.867 | 0.782 | 0.808 |
| CRNN_RES | 0.969 | 0.8985 | 0.8363 | 0.8296 |

**TABLE 3. COMPARISON OF MODEL SIZE, NUMBER OF PARAMETERS, AND RECOGNITION SPEED SETS**

| Network | Model size | Recognition speed | number of parameters |
|---|---|---|---|
| Dimension | | | |
| CRNN | 32m | 6.93ms | 7148325 |
| CRNN_RES | 28m | 6.07ms | 8331301 |

As can be seen from the table 2 and table 3, the method proposed in this paper is 1.40%, 3.15%, 5.43%, and 2.16% higher than CRNN in the four datasets, respectively. Compared with CRNN, the method proposed in this paper has significantly improved the accuracy. The network model proposed in this paper is 4M smaller than CRNN. The average recognition speed was 14% faster than that of CRNN. The CRNN_RES network proposed in this paper achieves a higher recognition accuracy than CRNN under the condition of faster speed and smaller models than CRNN.

## ACKNOWLEDGMENT

The authors would like to thank Dr. Chengjun Li, from Mogo Corporation for their insightful comments, thanks are due to Ming Yang for assistance with the experiments and valuable discussion, which have substantially improved the quality of this paper.

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## REFERENCES

1 Rumelhart, H. Learning representations by back-propagating errors. (1985).
2 LeCun. Gradient-Based Learning Applied to Document Recognition. (1988).
3 Dan Claudiu Cires, U. M., Luca Maria Gambardella, J¨urgen Schmidhuber. Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition. (2010).
4 Alex Krizhevsky, I. S., Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. (2012).
5 Karen Simonyan, A. Z. Very Deep Convolutional Networks for Large-Scale Image Recognition. (2014).
6 Kaiming He , X. Z., Shaoqing Ren ,Jian Sun. Deep Residual Learning for Image Recognition. (2015).
7 Baoguang Shi, X. B., Cong Yao. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. (2015).
8 A. Graves, M. L., S. Fernandez, R. Bertolami, & H. Bunke, a. J. S. Learning Precise Timing with LSTM Recurrent Networks. (2013).
9 F. A. Gers, N. N. S., and J. Schmidhuber. Long Short-Term Memory. (2002).
10 Ioffe, S. & Szegedy, C. in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* 448–456 (JMLR.org, Lille, France, 2015).
11 Max Jaderberg, K. S., Andrea Vedaldi, Andrew Zisserman. Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. (2014).

12 Lucas, S. M. *et al.* ICDAR 2003 robust reading competitions: entries, results, and future directions. *International Journal of Document Analysis and Recognition (IJDAR)* **7**, 105-122, doi:10.1007/s10032-004-0134-3 (2005).

13 D. Karatzas, F. S., S. Uchida, M. Iwamura, L. G. i Bigorda,, S. R. Mestre, J. M., D. F. Mota, J. Almazan, and ´ & Heras., L. d. l. in *2013 12th International Conference on Document Analysis and Recognition* 1484-1493 (2013).

14 A. Mishra, K. A., and C. V. Jawahar. Scene Text Recognition using Higher Order Language Priors. (2012).

15 K. Wang, B. B., and S. Belongie. End-to-End Scene Text Recognition. (2011).

# Figures

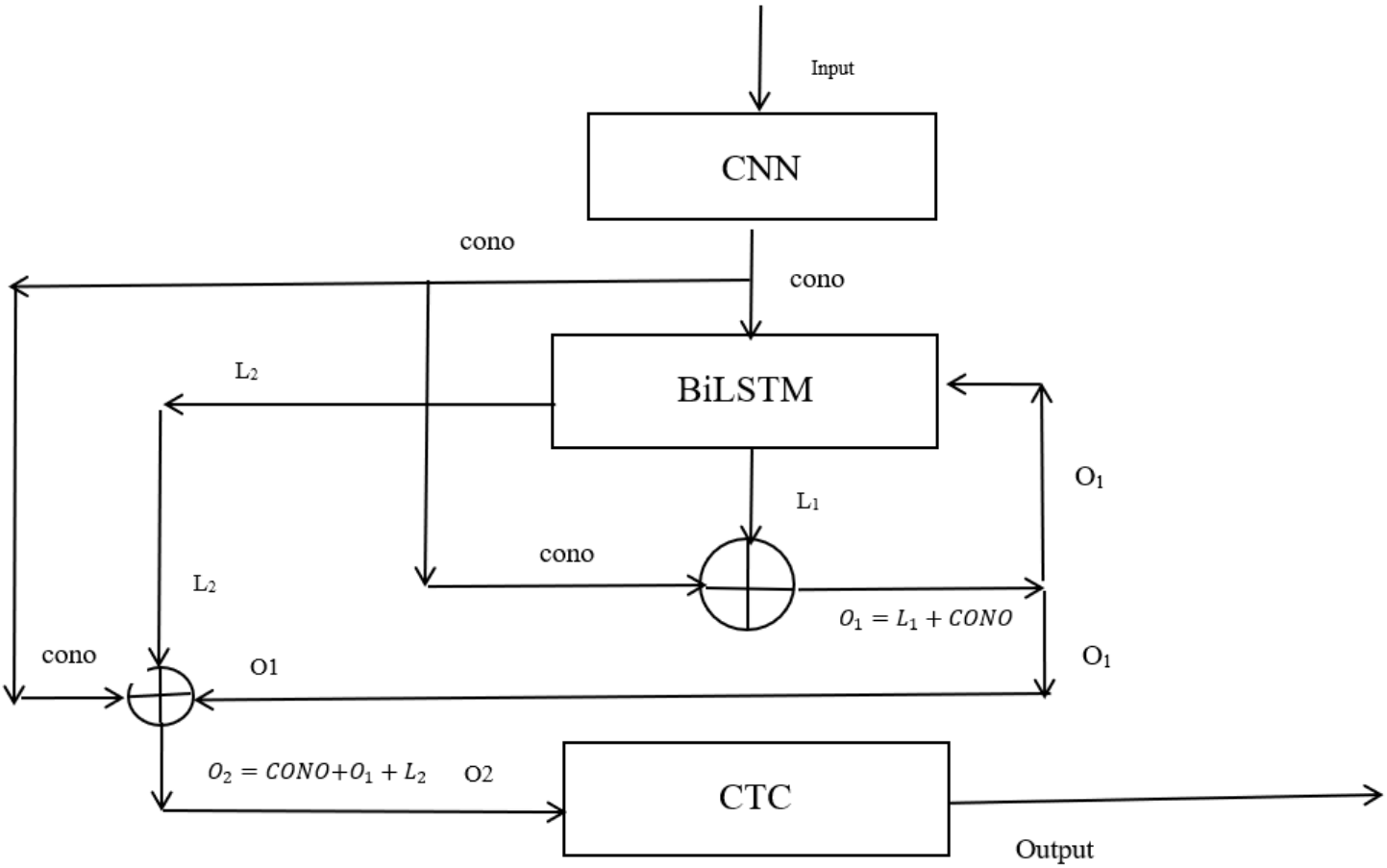
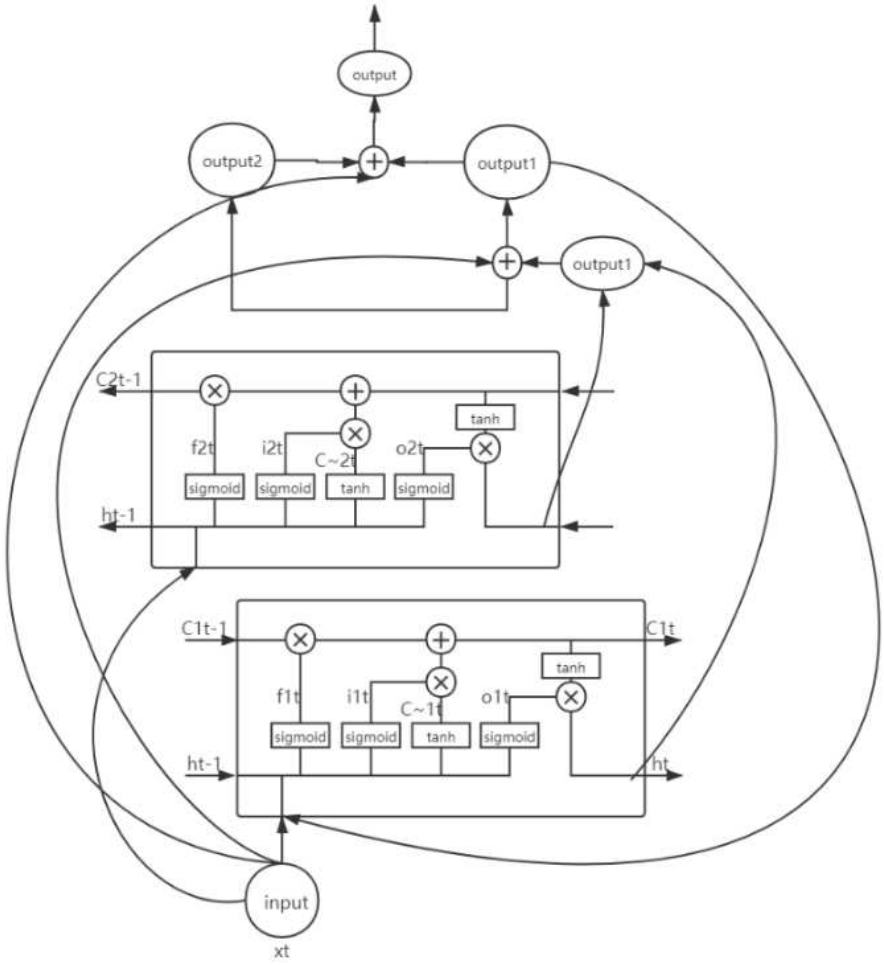



**Figure 1**

Summary architecture diagram of the RNN layer in the CRNN_RES network.




## Figure 2

Detailed architecture diagram of the RNN layer in the CRNN_RES network

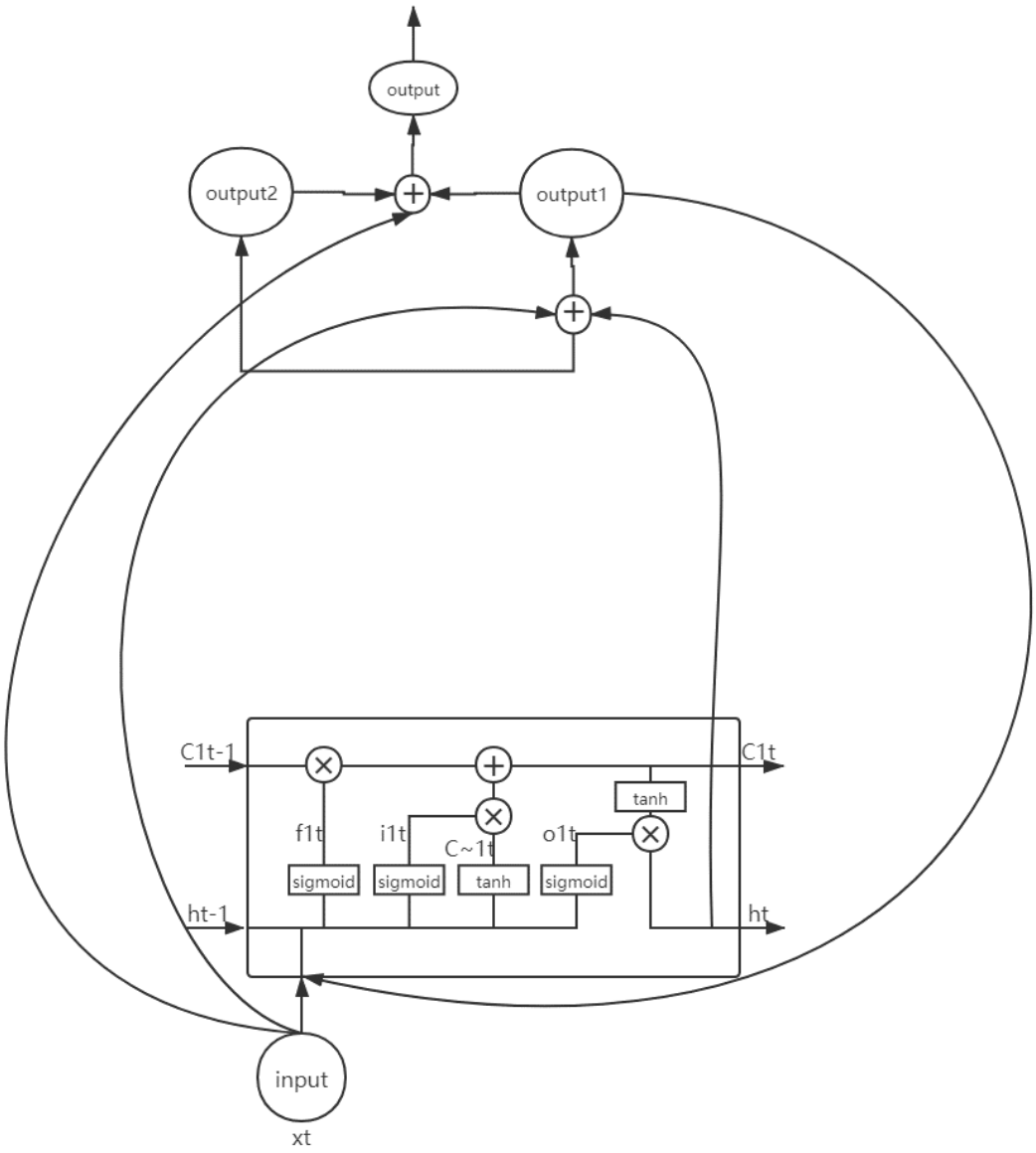



**Figure 3**

the form of one-way LSTM.

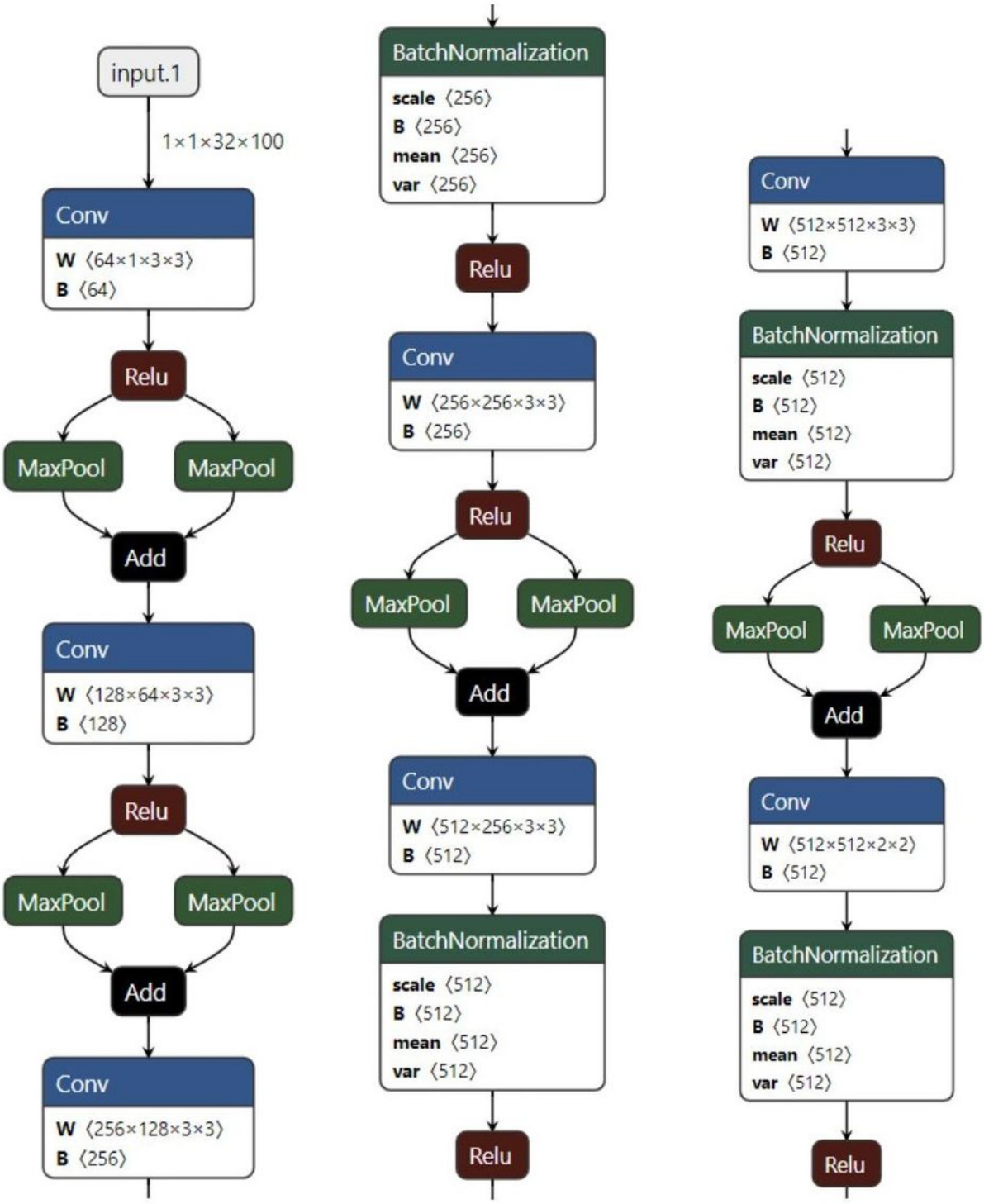



**Figure 4**

The CNN layer architecture of CRNN_RES.